AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing Of Claims:

Please amend the claims as follows:

1.-25. (Canceled)

26.   (Currently Amended) A system for collaboratively developing a computer application software product, the system comprising:

a first computer and a second computer linked by way of a network;

a first set of one or more computer programs embodied in a computer-readable medium used by a first system user operating the first computer for producing an executable program component of the computer application software product;

at least one user interface resource file embodied in a computer-readable medium, the at least one user interface resource file comprising a document in a markup language, wherein tagged text elements are associated with attributes of a user interface component of the computer application software product and the markup language includes a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes, the precedence order of default values comprising first using an attribute value associated with a previous occurrence of a tag in a tree if the tag is not provided with an overriding attribute, then using a default attribute value if no previous occurrence of the tag is present in the tree, if an acceptable attribute value is not obtained by the attribute value

associated with the previous occurrence of the tag in the tree or the default attribute value, then using an attribute provided by one of the application or the user wherein the user interface resource ~~data~~ file includes tags not specified in a schema or a document type definition if the acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value wherein the acceptable attribute value not being obtained by the attribute value associated with the previous occurrence of the tag comprises the attribute value associated with the previous occurrence of the tag causing overlapping controls; and

a second set of one or more computer programs embodied in a computer-readable medium used by a second system user operating the second computer for creating and modifying the user interface component by manipulating the at least one user interface resource file, wherein the creating and modifying by the second system user are independent of actions taken by the first system user.

27.   (Previously Presented) The system of claim 26 wherein the second set of one or more computer programs is used while the computer application software product is being executed, the creating and modifying occurring dynamically, and not requiring a recompilation of the executable program component.

28.   (Previously Presented) The system of claim 26, further comprising a set of operating system resource-loading routines embodied in a computer-readable medium for presenting a user interface corresponding to the user interface component to a third system user.

29.　(Previously Presented) The system of claim 28 wherein the resource-loading routines obtain user interface resource information from a user interface attribute data tree corresponding to the user interface resource file and, with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information.

30.　(Previously Presented) The system of claim 28 wherein the third system user is also the first system user.

31.　(Previously Presented) The system of claim 28 wherein the third system user is also the second system user.

32.　(Previously Presented) The system of claim 28 wherein the third system user is neither the first system user nor the second system user.

33.　(Canceled)

34.　(Currently Amended) A system for customizing a user interface for an executable computer program by a user of the program, the system comprising:

a computer operated by the user, the computer including a memory in which the executable computer program is stored;

at least one user interface resource file embodied in a computer-readable medium, comprising a document in a markup language, wherein tagged text elements are associated with attributes of the user interface and the markup language includes a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes, the precedence order of default values comprising first using an attribute value associated with a previous occurrence of a tag in a tree if the tag is not provided with an overriding attribute, then using a default attribute value if no previous occurrence of the tag is present in the tree, if an acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value, then using an attribute provided by one of the application or the user wherein the <u>user interface</u> resource ~~data~~ file includes tags not specified in a schema or a document type definition if the acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value <u>wherein the acceptable attribute value not being obtained by the attribute value associated with the previous occurrence of the tag comprises the attribute value associated with the previous occurrence of the tag causing overlapping controls</u>; and

a set of one or more computer programs, embodied in a computer-readable medium, for modifying the at least one user interface resource file.

35.   (Previously Presented) The system of claim 34 wherein the set of one or more computer programs for modifying the at least one user interface resource file is invoked while the computer program is being executed, the customizing occurring dynamically.

36.   (Previously Presented) The system of claim 34, further comprising a set of operating system resource-loading routines embodied in a computer-readable medium for presenting the user interface to the user, wherein the resource-loading routines obtain user interface resource information from a user interface attribute data tree corresponding to the user interface resource file and, with respect to resource information not specified in the user interface resource file, from a set of default resources of user interface resource information.

37.   (Canceled)

38.   (Currently Amended) A method for collaboratively developing a computer application software product by at least two system users, the computer application software product including a user interface, the method comprising:

by a first system user, writing source code for the computer application software product, and generating a first build of the computer application software product; and

by a second system user,

executing the first build, thereby causing the user interface to be presented;

proposing changes to the user interface;

if the proposed changes require the first system user to rewrite the source code and generate a second build, communicating the proposed changes to the first system user; and

if the proposed changes do not require the first system user to rewrite the source code and generate a second build, editing at least one user interface resource file to incorporate the proposed changes, each user interface resource file comprising symbols of a markup language having a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of a default values for the attributes, the precedence order of default values comprising first using an attribute value associated with a previous occurrence of a tag in a tree if the tag is not provided with an overriding attribute, then using a default attribute value if no previous occurrence of the tag is present in the tree, if an acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value, then using an attribute provided by one of the application or the user wherein the user interface resource ~~data~~ file includes tags not specified in a schema or a document type definition if the acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value, and causing a new user interface to be presented wherein the acceptable attribute value not being obtained by the attribute value associated with the previous occurrence of the tag comprises the attribute value associated with the previous occurrence of the tag causing overlapping controls.

39. (Previously Presented) The method of claim 38 wherein the at least one user interface resource file comprises a document in a markup language, wherein tagged text elements are associated with attributes of the user interface.

40. (Previously Presented) The method of claim 38 wherein the causing the user interface to be presented comprises:

parsing the at least one user interface resource file into a user interface attribute data tree;

invoking operating system resource-loading routines for constructing the user interface; and

obtaining user interface resource information from the user interface attribute data tree and, with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information.

41. (Previously Presented) The method of claim 38 wherein the causing the user interface to be presented occurs while the first build is being executed and does not require the first build to be re-executed.

42. (Currently Amended) A computer-readable medium storing computer-executable instructions for performing a method for collaboratively developing a computer application software product having a user interface, the method comprising:

executing a first build of the computer application software product;

opening at least one user interface resource file specifying characteristics of the

user interface, so that the at least one user interface resource file may be edited by a

user to effect changes to the user interface, each user interface resource file comprising

symbols of a markup language having a plurality of resource tag attribute default value

mechanisms whose order corresponds to a precedence order of default values for the

attributes, the precedence order of default values comprising first using an attribute

value associated with a previous occurrence of a tag in a tree if the tag is not provided

with an overriding attribute, then using a default attribute value if no previous

occurrence of the tag is present in the tree, if an acceptable attribute value is not

obtained by the attribute value associated with the previous occurrence of the tag in the

tree or the default attribute value, then using an attribute provided by one of the

application or the user wherein the user interface resource data file includes tags not

specified in a schema or a document type definition if the acceptable attribute value is

not obtained by the attribute value associated with the previous occurrence of the tag in

the tree or the default attribute value wherein the acceptable attribute value not being

obtained by the attribute value associated with the previous occurrence of the tag

comprises the attribute value associated with the previous occurrence of the tag

causing overlapping controls; and

causing a new user interface incorporating the changes to be presented, without

requiring a recompilation of the computer application software product.

43.   (Currently Amended) A method for customizing a user interface for an executable computer program by a user of the program, the method comprising:

executing the computer program, thereby causing the user interface to be presented;

editing at least one user interface resource file, the at least one user interface resource file comprising a document in a markup language, wherein tagged text elements are associated with attributes of the user interface and the markup language includes a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes, the precedence order of default values comprising first using an attribute value associated with a previous occurrence of a tag in a tree if the tag is not provided with an overriding attribute, then using a default attribute value if no previous occurrence of the tag is present in the tree, if an acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value, then using an attribute provided by one of the application or the user wherein the user interface resource data file includes tags not specified in a schema or a document type definition if the acceptable attribute value is not obtained by the attribute value associated with the previous occurrence of the tag in the tree or the default attribute value wherein the acceptable attribute value not being obtained by the attribute value associated with the previous occurrence of the tag comprises the attribute value associated with the previous occurrence of the tag causing overlapping controls; and

causing a new user interface to be presented.

44.   (Previously Presented) The method of claim 43, wherein the causing the user interface to be presented comprises:

parsing the at least one user interface resource file into a user interface attribute data tree;

invoking operating system resource-loading routines for constructing the user interface; and

obtaining user interface resource information from the user interface attribute data tree and, with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information.

45.   (Previously Presented) The method of claim 43 wherein the causing the user interface to be presented occurs while the computer program is being executed and does not require the computer program to be re-executed.

46.   (Previously Presented) A computer-readable medium storing computer-executable instructions and computer-readable data for performing the method of claim 43.

47.   (Previously Presented) The system of claim 34, wherein:

a parse tree is generated for the markup language of the document of each user interface resource file; and

the ordered plurality of resource tag attribute default value mechanisms comprises a preceding control default value mechanism that enables at least one

attribute value of a first resource tag to be derived from at least one attribute value of a like resource tag situated earlier in the parse tree.

48.   (Previously Presented) The system of claim 47, wherein the ordered plurality of resource tag attribute default value mechanisms further comprises a parent node default value mechanism that enables the at least one attribute value of the first resource tag to be derived from at least one attribute value of a parent resource tag that is a parent of the first resource tag in the parse tree.

49.   (Previously Presented) The system of claim 48, wherein:

the markup language of the document of each user interface resource file is interpreted in accordance with a schema; and

the ordered plurality of resource tag attribute default value mechanisms further comprises a style specification default value mechanism that enables the at least one attribute value of the first resource tag to be derived from a default style specification in the schema.

50.   (Previously Presented) The system of claim 49, wherein the preceding control default value mechanism takes precedence over the style specification default value mechanism.

51.   (Previously Presented) The medium of claim 42, wherein:

a parse tree is generated for the markup language of each user interface resource file; and

the ordered plurality of resource tag attribute default value mechanisms comprises a preceding control default value mechanism that enables at least one attribute value of a first resource tag to be derived from at least one attribute value of a like resource tag situated earlier in the parse tree.

52.   (Previously Presented) The medium of claim 51, wherein the ordered plurality of resource tag attribute default value mechanisms further comprises a parent node default value mechanism that enables the at least one attribute value of the first resource tag to be derived from at least one attribute value of a parent resource tag that is a parent of the first resource tag in the parse tree.

53.   (Previously Presented) The medium of claim 52, wherein:

the markup language of each user interface resource file is interpreted in accordance with a schema; and

the ordered plurality of resource tag attribute default value mechanisms further comprises a style specification default value mechanism that enables the at least one attribute value of the first resource tag to be derived from a default style specification in the schema.